LLNL-CONF-569332

Extending the MC-SURE to Denoise Sensor Data Streams

M. Ndoye, C. Kamath

August 3, 2012

46th Annual Asilomar Conference on Signals, Systems, and Computers
Pacific Grove, CA, United States
November 4, 2012 through November 7, 2012

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Extending the MC-SURE to Denoise Sensor Data Streams

Mandoye Ndoye and Chandrika Kamath
Lawrence Livermore National Laboratory
Livermore, California 94550
Email: ndoye1,kamath2@llnl.gov

***Abstract*—A signal processing method is proposed to adaptively denoise sensor data streams corrupted by noise that can be approximated as additive white Gaussian. This on-line filtering method takes advantage of the Monte-Carlo Stein's Unbiased Risk Estimate (MC-SURE) algorithm, which enables a blind optimization of the denoising parameters for a wide class of filters. We formulate a straightforward and practical framework for adequately denoising real sensor streaming data by identifying and solving the challenges that arise as the MC-SURE algorithm is applied in on-line data processing settings. The proposed technique has been successfully tested using real-world datasets.**

I. Introduction

The increasing affordability of sensors is enabling cost-effective and real (or near-real) time monitoring of complex phenomena and systems such as fusion in Tokamak reactors, electric power grids or large-scale infrastructure networks. In these systems, data analysis algorithms are applied to measurements captured from multiple sensors to monitor the phenomena or systems under consideration by tracking their operating states, detecting deviations from their normal behaviors and etc. The effectiveness of such real-time system monitoring algorithms depends, however, on the quality of the input data; i.e., their signal-to-noise power ratio. Since sensor data are typically subject to non-negligible measurement errors due to noise, recorded measurements must be adequately filtered to remove most of the noise contributions while preserving the important waveform information. Although a host of denoising algorithms already exist, most are not inherently designed for real (or near-real) time data processing: they often lack an automated mechanism for selecting the best parameter value for denoising arbitrary measurement sequences using the prescribed filter.

We develop a simple and practical method for on-line denoising of sensor data streams with arbitrary waveform characteristics by using the Monte-Carlo *Stein's Unbiased Risk Estimate* (MC-SURE) algorithm [1]: an approach that enables a blind optimization of the regularization parameters of a wide class of filters that seek to recover an arbitrary signal corrupted by additive white Gaussian noise (AWGN). The MC-SURE formulation is particularly suited for the denoising of sensor data streams since it produces the *mean-squared-error*-based optimal denoising parameter for a chosen filter without any assumption about the underlying noise-free signal. This work concerns the development of a practical strategy for enabling effective sensor data streams by identifying then subsequently addressing all the implementation issues that emerge while applying the MC-SURE algorithmic concept to on-line data processing settings. The paper is organized as follows. In Section II, the general concept of the MC-SURE algorithm is described and the associated on-line-implementation issues are discussed. Section III deals with the steps taken to address the encountered problems. Results and discussions are given in Section IV and concluding remarks, in Section V.

II. Concepts and Challenges

The MC-SURE algorithm optimizes the (vector)-parameter λ of a continuous and (weakly) differentiable denoising function $f_\lambda(\cdot)$. Consider the noisy data

$$\mathbf{y} = \mathbf{s} + \mathbf{w} \tag{1}$$

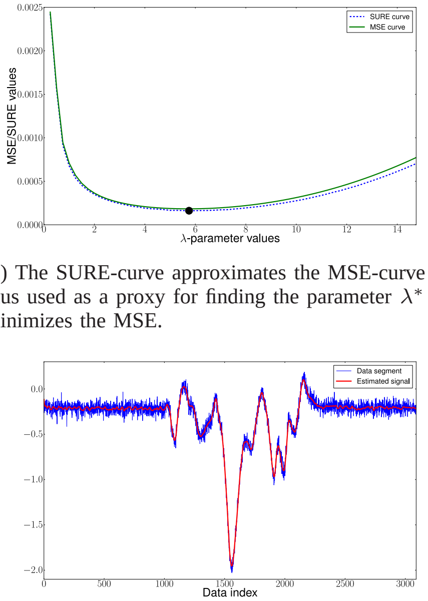
comprised of a desired signal $\mathbf{s} \in \mathbb{R}^N$ and $\mathbf{w} \in \mathbb{R}^N$, a zero-mean AWGN with variance σ^2. The mapping

$$\hat{\mathbf{s}}_\lambda = f_\lambda(\mathbf{y}) \tag{2}$$

is the λ-parametrized data filtering operation that produces a signal estimate of $\mathbf{s}$. The MC-SURE procedure finds the optimal parameter for denoising $\mathbf{y}$ by minimizing a proxy of the mean-squared-error (MSE) criterion; namely, the *Stein's Unbiased Risk Estimate* (SURE) [2]. This SURE-statistic estimating the MSE is expressible [1] as

$$T_\lambda(\mathbf{y}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{s}}_\lambda\|^2 + \frac{2\sigma^2}{N}\text{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\} - \sigma^2 \tag{3}$$

where $\text{div}_{\mathbf{y}}\{f_\lambda(\mathbf{y})\}$ denotes the divergence of the denoising function with respect to the data. The MSE-based optimal denoising parameter λ^* is thus the minimizer of the SURE-curve $\{T_\lambda(\mathbf{y}) : \lambda \in \mathbb{R}^K\}$. Fig. 1 illustrates the automated denoising of a data segment using the MC-SURE. The MC-SURE procedure is particularly well-suited to online data denoising as it produces the *best* filter parameter without any assumption about the underlying signal. Furthermore, using a Monte-Carlo approach to compute the divergence term in Equation 3, provides more options when selecting a noise-removal filter since we are now not limited to filters with closed-form expressions. The application of the MC-SURE algorithm to streaming data presents however some practical challenges. In online data processing settings, the need to minimize latency dictates that processed data blocks have a



(a) The SURE-curve approximates the MSE-curve and thus used as a proxy for finding the parameter λ^* that minimizes the MSE.

(b) The data and the estimated signal obtained by filtering the data with the Gaussian filter with parameter $\lambda = \lambda^*$.

Fig. 1: Illustration of the MC-SURE procedure.

limited number of observations. This requirement introduces some performance issues, which must be properly addressed to arrive at an effective online strategy. These issues are the following.

Noise estimation errors: The SURE-statistic formula from Equation 3 assumes the noise term σ is exactly known. However, it has to be estimated in practice. In [1], Ramani *et al.* replaces σ by its estimate computing using the Donoho median estimator method [3] that approximates the noise standard deviation of a length-N dataset $\mathbf{y}$ by $\hat{\sigma} = M_{\mathbf{y}}/0.6745$ where $M_{\mathbf{y}}$ is the median of the $N/2$ wavelet coefficients at the finest scale. In the above case, substituting σ by $\hat{\sigma}$ is reasonable because the noise is exactly white Gaussian and the number N of observations is very large (N $\geq$ 256^2): [the analysis in [1] was done on images with synthetically added AWGN]. However, in streaming/online data processing settings, the size of a data block will be relatively small. As a result, errors in estimating the noise term will be significant and will lead to unreliable SURE-curves. The length-N segments in Fig. 2 are from the same periodic signal and thus have in principle the same frequency content. When N is equal to 25, more substantial discrepancies are observed between SURE-curves as compared to the case where N equal 300. This can be attributed to the increased variance in estimating the noise as N gets smaller.

Errors in divergence-term computation: Given a length-N data segment $\mathbf{y}$ and a set of candidate denoising parameter values $\{\lambda_j : j = 1, 2, \ldots, J\}$, the reliability of the computed SURE-statistics $T_{\lambda_j}(\mathbf{y})$ also depends on the accuracy of the computed divergence term $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$. The divergence can

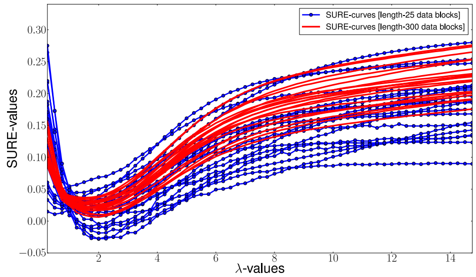


Fig. 2: SURE-curves for two groups of data segments: 20 length-25 data segments and 20 length-300 data segments

be expressed as

$$\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\} = \lim_{\epsilon \to 0} E_{\mathbf{b}}\{\mathbf{b}^T(f_{\lambda_j}(\mathbf{y}+\mathbf{b}) - f_{\lambda_j}(\mathbf{y}))\}, \quad (4)$$

where $\mathbf{b}$ is a zero-mean i.i.d. random vector with covariance $\epsilon^2\mathbf{I}$, and $E_{\mathbf{b}}\{\cdot\}$ denotes expectation with respect to $\mathbf{b}$. When a proper value of ϵ is chosen, $\text{div}_{\mathbf{y}}\{f_{\lambda_j}(\mathbf{y})\}$ is estimated using a Monte-Carlo (MC) approach: 1) Generate k realizations $\{\mathbf{b}_i\}_{i=1}^k$ of a length-N random vector $\mathbf{b}$ and 2) compute the k-MC-run divergence estimate

$$\widehat{\text{div}}_{\mathbf{y}}^{(k)}\{f_{\lambda_j}(\mathbf{y})\} = \frac{1}{\epsilon k}\sum_{i=1}^{k}\mathbf{b}_i^T(f_{\lambda_j}(\mathbf{y}+\mathbf{b}_i) - f_{\lambda_j}(\mathbf{y})), \quad (5)$$

which amounts to averaging k single-MC-run divergence estimates. A single MC-run suffices to generate a reliable estimate of divergence if N is large. The estimation error increases rapidly however with decreasing N as shown in Fig. 3. Thus, when N is relatively small (e.g., online block-based data processing), estimates from a single run are unreliable as shown in Fig. 4. In such situations, a proper number k of MC-runs must be found on-the-fly to reliably compute the divergence-terms for each data segment.

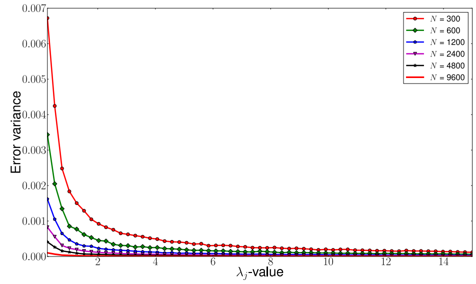


Fig. 3: Variance of the error in estimating the divergence-curve for different values of the data size N.

Detrimental effects of strong DC components: Numerical experiments suggested that the MC-SURE procedure fails to return an adequate denoising parameter for data with a strong DC-component. This issue is independent of the length of the data block under consideration. Fig. 5 illustrates the point. The data segments $\mathbf{z}$ and $\mathbf{z}_D$ are identical except for the

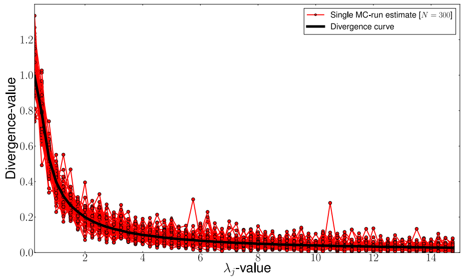
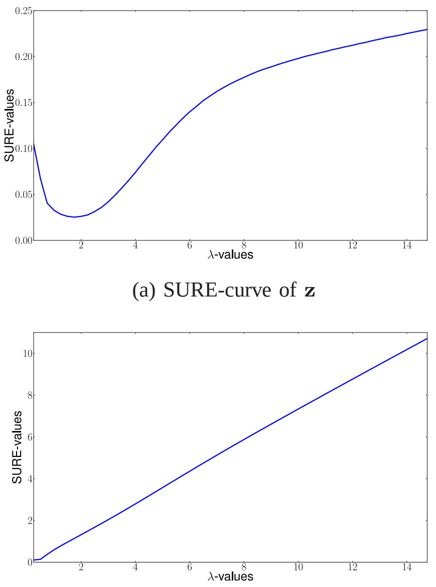


Fig. 4: Comparison of the divergence-curve with single-MC-run estimate when the data size is relatively low: $N = 300$ samples.

presence of a strong DC-component: $\mathbf{z}_D = \mathbf{D} + \mathbf{z}$, where $\mathbf{D}$ is a constant-vector. The application of the MC-SURE algorithm to $\mathbf{z}$ produces the SURE-curve in Fig. 5a, which reports the adequate filter parameter value of $\lambda^* = 2$. Since $\mathbf{z}_D$ and $\mathbf{z}$ are identical except for the presence of a DC term, a similar denoising parameter value should be needed for $\mathbf{z}_D$. The corresponding SURE-curve, shown in Fig. 5b, erroneously reports however that no denoising is needed ($\lambda^* = 0$). In general, when the data under consideration has a very strong DC component, the MC-SURE algorithm appears to always report that no denoising is needed regardless of the amount of noise present.



(a) SURE-curve of $\mathbf{z}$

(b) SURE-curve of $\mathbf{z}_D$ ($\mathbf{z}$ plus DC-component)

Fig. 5: Comparison of the SURE-curves generated using the datasets $\mathbf{z}$ and $\mathbf{z}_D$. Although the two SURE-curves differ significantly, $\mathbf{z}$ and $\mathbf{z}_D$ only differ by a DC-component term.

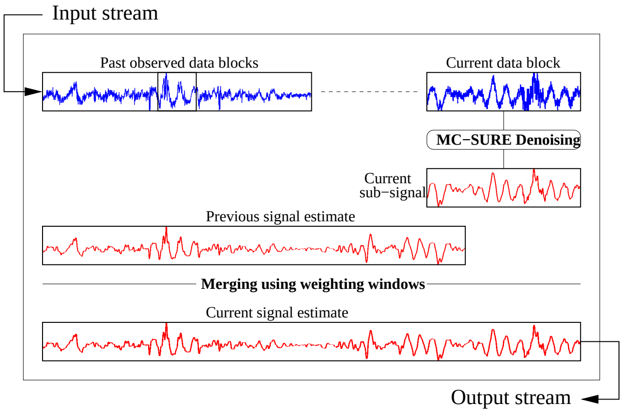


Fig. 6: Schematic system representation of the block-based on-line data processing framework centered on the MC-SURE algorithm.

III. A Practical Strategy for Denoising Sensor Data Streams

Consider a sensor data stream $\mathbf{y} = \{y_n : n = 0, 1, \dots\}$ comprised of a desired signal $\mathbf{s} = \{s_n : n = 0, 1, \dots\}$ plus AWGN. The model assumes that the noise parameter varies as time elapses, albeit very slowly. Samples from $\mathbf{y}$ are taken in as successive data blocks of properly chosen sizes. The i-th data block is denoted by $\mathbf{y}_i = \{y_n : N_i \leq n < M_i\}$, where $N_{i+1} = M_i - L$ and L is the overlap between consecutive data blocks. The stream of data blocks $\{\mathbf{y}_i : i = 0, 1, \dots\}$ is denoised one at a time. The noise in the i-th data block $\mathbf{y}_i$ is filtered out using the prescribed filter $f_\lambda(\cdot)$ with parameter λ set to λ_j^*, the best MC-SURE-based parameter value. The i-th outputted data block $\hat{\mathbf{s}}_i = f_{\lambda_j^*}(\mathbf{y}_i)$ is the sub-signal of $\mathbf{s}$ contained in $\mathbf{y}_i$. The estimate of the portion of $\mathbf{s}$ that is available after processing the ($\mathcal{I}$+1)-th data block $\mathbf{y}_{\mathcal{I}+1}$ is denoted $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$. This current signal estimate $\hat{\mathbf{s}}^{(\mathcal{I}+1)}$ is formed by stitching together the available sub-signal estimates: $\{\hat{\mathbf{s}}_0, \hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_{\mathcal{I}+1}\}$. This is done recursively via

$$\mathbf{s}^{(\mathcal{I}+1)} = \mathbf{s}^{(\mathcal{I})} \odot \mathbf{W}^{(\mathcal{I})} + \hat{\mathbf{s}}_{\mathcal{I}+1} \odot \mathbf{W}_{\mathcal{I}+1}, \tag{6}$$

where $\mathbf{s}^{(\mathcal{I}+1)}$ is new signal-estimate constructed by properly merging via tapered windows the newly obtained sub-signal estimate $\hat{\mathbf{s}}_{\mathcal{I}+1}$ with the previous signal-estimate $\mathbf{s}^{(\mathcal{I})}$. The operator $\odot$ denotes element-by-element multiplication. The terms $\mathbf{W}^{(\mathcal{I})}$ and $\mathbf{W}_{\mathcal{I}+1}$ denote here one-sided cosine tapering windows although other types of tapering windows are also usable. The recursive procedure is initialized with $\mathbf{s}^{(0)} = \hat{\mathbf{s}}_0$. The above data processing strategy is illustrated in Fig. 6. For this system to be effective in denoising streaming data, individual data blocks must be properly denoised using the MC-SURE in spite of their relatively short lengths or the presence of DC terms. The steps taken to address the previously mentioned issues are described below.

Removal of DC components: To suppress the detrimental effects of DC components on the MC-SURE's performance, we first pass each data block $\mathbf{y}_i$ through a notch-filter centered at the DC-frequency since a data block without a

DC-component will be indifferent to this pre-filtering. For simplicity, we may assume hereon that each data block is free of a DC-component: it would already be removed at this stage, if present.

Noise estimation: We previously noted that estimating the noise term σ using only the limited number of samples from a data block $\mathbf{y}_i$ produces an inaccurate estimate which, in turn, deteriorates the accuracy of the computed SURE-curve. We mitigate the problem by combining observations from a chosen number of consecutive streaming data blocks to better estimate σ. Since noise characteristics are expected to change slowly with time, it is reasonable to assume noise observations from M consecutive data blocks come from i.i.d. Gaussian distributions with parameter σ. The Donoho median estimator produces a separate estimate of σ from each of the M previous data blocks. When processing the current data block $\mathbf{y}_\mathcal{I}$, the noise estimates from the M previous data blocks $\{\hat{\sigma}_{\mathcal{I}-M+1}, \hat{\sigma}_{\mathcal{I}-M+2}, \dots, \hat{\sigma}_\mathcal{I}\}$ are combined to produce a more reliable estimate of σ via

$$\bar{\sigma}_\mathcal{I} = \frac{\sum_{i=\mathcal{I}}^{\mathcal{I}-M+1} N_i \hat{\sigma}_i}{\sum_{i=\mathcal{I}}^{\mathcal{I}-M+1} N_i} \tag{7}$$

where N_i denotes the length of the i-th data block. The weighted averaging takes into consideration the unequal reliabilities of individual noise estimates and gives more weight to those from longer data blocks, as they are expected to be more reliable.

Divergence-curve computation: For the grid of parameter-values $\{\lambda_j : j = 1, 2, \dots, J\}$, the divergence-curve is estimated via Equation 5 using k MC runs is denoted by

$$d_j^{(k)} = \widehat{\mathrm{div}}_{\mathbf{y}_\mathcal{I}}^{(k)}\{f_{\lambda_j}(\mathbf{y}_\mathcal{I})\},\ j = 1, 2, \dots, J \tag{8}$$

For a fixed j, $\{d_j^{(k)} : k = 1, 2, \dots\}$ is a converging sequence of estimates of $\mathrm{div}_\mathbf{y}\{f_{\lambda_j}\}$. The j-th element of the divergence-curve is thus estimated by $d_j^{(K_j)}$ with the number of MC-runs K_j chosen such that

$$r_j^{(K_j)} = \frac{|d_j^{(K_j+1)} - d_j^{(K_j)}|}{|d_j^{(2)} - d_j^{(1)}|} < \delta, \tag{9}$$

and where δ is the convergence stopping criterion. Seeking K_j for each of the J divergence-values may be impractical: The convergence rate of $\{r_j^{(k)} : k = 1, 2, \dots\}$ differ from one j-index to another. Thus, instances might occur where, for one or a few j-indexes, an unreasonably large number of runs is needed to satisfy Equation 9. A group convergence requirement is then used to avoid such situations: For all parameter values λ_j, the divergence-value is estimated by $d_j^{(K)}$ with the number of MC-runs chosen K such that

$$r^{(K)} = \frac{1}{J}\sum_{j=1}^{J} r_j^{(K)} < \delta. \tag{10}$$

The k-indexed sequence $r^{(k)}$ goes from one to zero as the divergence-curve estimate $\{d_j^{(k)} : j = 1, 2, \dots, J\}$ converges with additional MC runs. Fig. 7 shows an example profile for $r^{(k)}$, where the choice of δ = 0.02 leads to the termination of the divergence-curve estimation after K = 67 MC simulations.

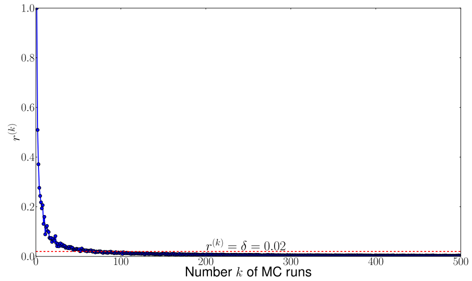


Fig. 7: Illustration of the profile of the sequence $r^{(k)}$, which is used to track the convergence of the divergence-curve estimate.

Online selection of data block size: To avoid potential spatial aliasing issues in the processing of the streaming data segments, sizes of data blocks are selected online using waveform structure information computed from available data. Techniques from scale-space theory [7] produce the typical length (e.g., scale) of the most salient signal structures in a dataset, and help choose the minimal data block length required to capture most of the relevant information. By choosing the size of the next block sufficiently larger than the data scale, enough samples are taken to avoid potential spatial aliasing issues. Since processing latency has to remain minimal, the sizes of windows must be bounded above. At the $\mathcal{I}$-th data block, the computed data scales for the last P data blocks are $\{\Delta_{\mathcal{I}-P+1}, \Delta_{\mathcal{I}-P+2}, \dots, \Delta_\mathcal{I}\}$. The size $N_{\mathcal{I}+1}$ of the ($\mathcal{I}$+1)-th data block is determined via

$$N_{\mathcal{I}+1} = \min\left\{N_{\max}, \theta \cdot \max\left\{\{\Delta_i\}_{i=\mathcal{I}-P+1}^{\mathcal{I}}\right\}\right\} \tag{11}$$

where θ is a multiplicative factor and $N_{\max}$ is the maximal data block size.

IV. Results and Discussions

TABLE I: Selected values for the needed parameters.

Parameter	Chosen value
Standard deviation of probing noise: ϵ	0.0002
Maximum allowed segment size: $N_{\max}$	500
Fixed overlap between segments: L	30
Multiplicative factor: θ	24
Number P of past blocks	2
Number M of past blocks	5

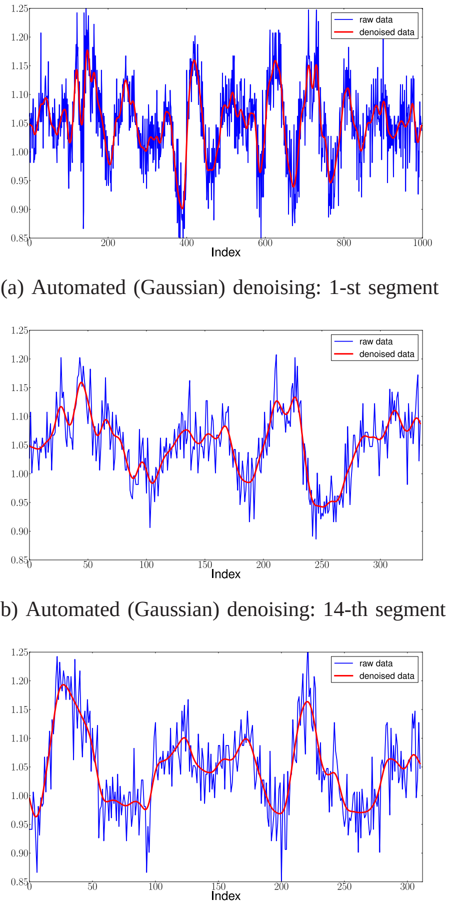
We tested the denoising strategy using synthetic and real datasets. Sample results and implementation steps are illustrated here with a real-world dataset: a sequence of vehicle body acceleration response measurements. Table I gives the parameter values selected in the implementation. We observed (as noted in [1]) that the MC-SURE is very robust to variations in the value of ϵ, which can change from 1 to 10^{-12} without discernible changes in performance. Since no previous data is available to systematically decide the initial data block size, it is chosen equal to $2N_{\max}$. The sizes of following blocks are systematically computed via Equation 11 using the largest of the scales of the two blocks preceding them (i.e., $P = 2$). Table II gives computed noise statistics and denoising parameters for 20 consecutive data blocks. The parameter $\bar{\sigma}_i$, used as noise term when computing the SURE-curve for the i-th data block, is the weighted average of the individual block noise estimates from the last five data blocks (i.e., $M = 5$). The denoising parameter λ_i^* in this case is the MC-SURE-chosen standard deviation of the Gaussian filter. Sample results of the sensor data streams denoising strategy are shown in Fig. 8. Such results support the premise that good performance is expected when the noise is AWGN or approximately so.

TABLE II: Computed statistics/parameters for 20 segments.

Data block	Scale	Length	$\hat{\sigma}_i$	$\bar{\sigma}_i$	λ_i^*
1	12	1000	0.0396	0.0396	3.75
2	12	244	0.0553	0.0427	4.25
3	15	244	0.0480	0.0435	7.0
4	19	360	0.0357	0.0420	5.0
5	13	456	0.0398	0.0416	3.25
6	13	456	0.0464	0.0440	4.25
7	11	312	0.0405	0.0419	3.25
8	15	312	0.0527	0.0429	3.25
9	12	360	0.0476	0.0451	3.75
10	13	360	0.0381	0.0451	3.25
11	31	312	0.0323	0.0423	3.75
12	13	500	0.0368	0.0411	2.75
13	14	500	0.0322	0.0372	3.0
14	13	336	0.0382	0.0355	3.5
15	13	336	0.0393	0.0357	3.5
16	15	312	0.0381	0.0366	1.0
17	13	360	0.0486	0.0388	0.75
18	11	360	0.0590	0.0450	0.5
19	12	312	0.0554	0.0483	3.75
20	14	288	0.0650	0.0533	0.5

V. Conclusion

We have proposed in this paper a practical strategy to automatically denoise sensor streaming data that have been corrupted by (approximately) Gaussian noise. First, we recognized that the MC-SURE algorithm is well-positioned to denoise sensor data streams since it prescribes a procedure for optimizing the regularization parameters of a wide class of denoising filters in a blind fashion; i.e., without any assumption about the underlying signal of interest. Second, we identified the implementation challenges that emanate when applying the MC-SURE algorithm to data blocks of limited size, which are to be expected in the scenario of streaming data processing. By introducing steps to resolve the identified issues, we arrive at a straightforward and practical strategy for effectively denoising sensor data streams. This adaptation of the MC-SURE provides flexibility when selecting a denoising filter in online data processing. Results from our tests on real datasets (that reasonably fit the data model) shows good denoising performance. Since the data model does not include datasets whose noise cannot be approximated by AWGN, a natural future research direction would be to extend this work to find an adaptation that would facilitate the handling of non-Gaussian noise.



(a) Automated (Gaussian) denoising: 1-st segment

(b) Automated (Gaussian) denoising: 14-th segment

(c) Automated (Gaussian) denoising: 7-th segment

Fig. 8: Illustration of denoising with the Gaussian filter.

Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] S. Ramani, T. Blu and M. Unser,*Monte-Carlo SURE: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms*, IEEE Transactions of Image Processing, Vol. 17, No. 19, September 2008.

[2] C. Stein, *Estimation of the Mean of a Multivariate Normal Distribution*, Annals of Statistics, Vol. 9, Pages 1135-1151, 1981.

[3] D. L. Donoho, *Adapting to Unknown Smoothness via Wavelet Shrinkage*, J. Amer. Statist. Assoc., Vol. 90, No. 432, Pages 1200-1224, 1995.

[4] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications. 4th Edition*, Prentice Hall, 2007.

[5] S. Orfanidis, *Introduction to Signal Processing*, Prentice Hall, 1995.

[6] A. P. Witkin, *Scale-space Filtering*, Proc. 8th Int. Joint Art. Intell., Karlsruhe, Germany, Pages 1019-1022, 1983.

[7] T. Lindeberg, *Scale Space Theory: A Basic Tool for Analyzing Structures at Different Scales*, Journal of Applied Statistics, Vol. 21, No. 2, Pages 224-270.